LoRa System for Monitoring and Facial Recognition

Vitor J. C. Rodrigues, Douglas F. Medeiros, Fabrício B. S. Carvalho, *IEEE Senior Member* and Waslon T. A. Lopes, *IEEE Senior Member*

***Abstract*—With the increasing tendency of incorporating technology into environments and activities in everyday life, new methods are being proposed to better integrate devices and mankind. Networks designed for monitoring areas through video and image systems are being implemented in several applications. Recent studies have shown interest from both the academy and industry to integrate wireless surveillance networks with low-cost and long range transmission technologies, such as LoRa (Long Range). With the intention to explore this topic, this work presents the development of a system prototype for intelligent monitoring as a basis to future implementation of low-cost LoRa-based Wireless Image Sensor Networks (WISN). The developed system consists of a processing unit, a data routing interface and multiple sensor nodes. Its operation is autonomous and initiated through the detection of human presence, by the sensor node, capturing an image of the individuals present at each detection. The data is transmitted through LoRa devices to the central hub, where python-based methods of facial detection and recognition are employed. Several tests were performed to both adjust the system and to verify its efficiency. The results achieved indicate the viability of the proposed low-cost LoRa-based WISN.**

***Index Terms*—LoRa; Wireless Image Sensor Networks; Digital Image Processing; Facial Recognition.**

I. Introduction

In its first appearance, in the late 1990s, the expression Internet of Things (IoT) was much more of a marketing term than an actual technological concept. The idea was that the growing automation market could solve the problems in industrial production caused by human factors. Nowadays, this concept has expanded to a much broader meaning and it can refer to any pervasive integration technology associated to a specific task in daily life [1], [2].

Recent days are witnessing an impressive growth in IoT applications. The amount of IoT devices in operation has already surpassed the world's population and estimates indicate that, by 2025, there will be over 21 billion connected devices worldwide [3].

There are different ways to integrate IoT devices. Wi-Fi, Bluetooth, Wireless Sensor Networks (in which the ZigBee protocol has a significant role) and cellular networks (GSM, 3G, 4G, 5G) are the most prominent technologies. This is due to the reduction in costs and complexity associated to employing such networks when compared to regular cable-based networks [4].

However, the increasing need of low-cost and low-power solutions in large-scale has revealed many restrictions related to signal range and power consumption regarding these technologies. Thus, different efforts were conducted to find out new technologies that best fit those characteristics. Devices with such features have since been classified as Low Power Wide Area Network (LPWAN). To guarantee a reasonable performance in the signal range without extending the power consumption, the main drawback that LPWAN devices present is a reduction in transmission rates. With a growth in popularity, it is expected that, by 2022, all the human population will have access to LPWAN coverage [4].

Among the most adopted LPWAN technologies, LoRa and SigFox are taking a significant role in LPWAN applications. Sigfox is more dedicated in offering long range, promising a nominal coverage of up to 40 km, although penalizing its data rate (limited up to 100 bps). On the other hand, the LoRa approach has a balance between coverage and data rate, with a data transmission rate in the order of kbps and a nominal range up to 20 km [5].

Nowadays, the majority of LPWAN-based applications involve Wireless Sensors Networks (WSN). Among them, a considerable number of applications is based on systems with reduced data stream between devices, which are, therefore, already adapted to the current context of LPWAN. Applications such as location and activity monitoring system [6]; an atmospheric pollutant monitoring system [7]; an acquisition system for a hydraulic plant [8]; a tracking and monitoring system for lightweight boats [9]; and a manhole cover monitoring device [10], among several others proposals published by researchers from all around the world. However, a second group of WSN, in opposition, requires more robust data flows to monitor/control their specific tasks, being much more impacted by current technological limitations - and therefore much more prone to push development forward and overcome those limitations [11], [12].

In the context of WSN, a new research and application field is arising: The Wireless Image Sensor Networks (WISN), frequently implemented in fields such as public and private security, environment monitoring, aerospace observation, among others. As those systems' primary source of information are images, usually characterized by a large volume of data, progress in this area is intrinsically associated with advancements in the data rate capabilities of LPWAN.

Despite the usual choice for Wi-Fi technology when deploying WISN applications today, an increasing number of scientific publications are now considering LoRa as a suitable technology for those systems, as shown in [13]–[18]. In these

Vitor J. C. Rodrigues, Douglas F. Medeiros, Fabrício B. S. Carvalho and Waslon T. A. Lopes are with the Communications and Signal Processing Research Group (GCOMPS) and the Post-Graduation Program in Electrical Engineering (PPGEE) at the Federal University of Paraíba (UFPB), João Pessoa-PB, Brazil. E-mails: vitor.rodrigues@cear.ufpb.br; douglas.medeiros@cear.ufpb.br; fabricio@cear.ufpb.br; waslon@cear.ufpb.br.

The authors would like to thank the Brazilian National Council for Scientific and Technological Development (CNPq) under Grant No. 315514/2018-3 and Coordination for the Improvement of Higher Education Personnel (CAPES) for the support to this research.

Digital Object Identifier: 10.14209/jcis.2021.1

works, the authors highlight the challenges to deploy such applications as a consequence of the current technology, whilst it is justified the advantages of the WISN technology that overcome the technical restrictions in LPWAN scenarios.

Considering the potential of LoRa-based WISN dedicated to monitoring, this work proposes the development of a system that consolidates the viability of LoRa as a solution for LPWAN with relatively high data rates. Additionally, this paper presents an alternative solution for image monitoring and/or biometry systems and demonstrates the potential to be expanded to larger and more complex sensor networks in that field.

As overall contributions provided by this work, the following items can be highlighted:

- A parsing technique for image files with HD and Full HD resolution to be sent through LoRa;
- A Machine-to-Machine (M2M) handling routine for routers receiving multiple transmissions at the same time;
- Experimental results on the behavior of LoRa transmissions under high data rate settings;
- A solution for surveillance and/or biometry based on face recognition and LoRa WSN.

Despite the contributions mentioned above, the system presented in this work finds itself in the position of an unique solution among its equivalents. Several available devices offer solutions in facial recognition, although the majority is either simple products designed for very specific day-to-day authentications (such as "1-second computer login webcams") and others are high-price robust authentication systems that might be too costly for most applications. There are also some rare examples of wireless camera systems, of which most are also some sort of smart home gadget, relying exclusively on Wi-Fi connection.

The proposed system, on the other hand, offers a Wireless LPWAN Network in the sense that it provides, even in a prototype version, an easy-access multi-nodal monitoring system, with no physical connection to the user and no cloud-driven commuting between nodes needed. Additionally, it offers a long range connection that allows a variety of monitoring applications in urban, rural and woodland environments, allowing even applications with moving nodes (such as "scouting networks", with nodes attached to people, animals or robots), since LoRa has a highly consistent transmission time throughout its coverage area.

Regarding the proposed system, it can be described as an intelligent wireless system for monitoring based on LoRa technology. This system is composed by three stages: a central processing unit, a router interface and one (or more) sensor node(s).

The operation of the proposed system is independent of human intervention and is composed of an idle/standby phase and an active phase, automatically triggered when a human presence is detected by a motion sensor. This starts a sequence encompassing the gathering, transmission and processing of an image, focused on detecting and recognizing the individuals that were detected. The wireless transmission of the data is completely based on LoRa and the facial recognition techniques implemented are based on Python language using robust artificial neural networks models. Experiments were carried out with the system's prototype to evaluate its overall performance as well as the viability of the LoRa technology as a technical and commercial solution for WISN.

In this context, this paper is organized as follows: LoRa technology and the other technologies used in the design of the proposed system - Passive Infrared (PIR), Complementary Metal Oxide Semiconductor (CMOS) image sensors, motion detectors and the chosen facial recognition Application Programming Interface (API)) - are described in Section II. With some technical background for the practical implementation, the system's prototype was designed utilizing both hardware and software elements, as detailed in Section III. After the development of the system, experimental tests were performed and the achieved results are highlighted in Section IV. At last, the final discussions related to this work are presented in Section V.

II. Background

This section briefly presents the concepts related to technologies used in the design of the proposed system. Section II.A introduces the fundamentals of LoRa technology and its characteristics. Section II.B describes the physics and the operation of the Pyroelectric Sensors (PIR). Section II.C presents the basic principles of Image Sensors based on CMOS technology. Lastly, Section II.D describes the Face Recognition API utilized and the methods on which it was based on.

A. LoRa

LoRa technology was developed in 2008 as an intellectual property of the French company Cycleo SAS, and later incorporated by Semtech, one of the founders of the LoRa Alliance. The LoRa Alliance is a non-profit organization devoted to the development and diffusion of LPWAN networks for IoT [19].

As a leading LPWAN tecnology, its robustness and long range in transmissions are important features for LoRa. A study carried out in dense urban environment showed that, even with many buildings, LoRa is able to transmit over distances in the order of kilometers with very low packet loss rate. In [20] the authors fixed the transmitter on the top of a building and moved the receiver to various points in the city of João Pessoa (Brazil), observing the Received Signal Strength Indication (RSSI) and the Packet Error Rate (PER) at each point. At the end, the results showed that the LoRa was able to transmit information to points located 4 km away with packet loss less than 2%. However, a high packet loss was also observed in points close to dense vegetation.

A similar study, carried out in an industrial environment considering short distances, evaluated the performance of LoRa in terms of packet loss and noise robustness. The authors used an industrial environment with many laser cutting machines to transmit data via LoRa, being the main objective of this study to demonstrate that it is reliably possible to monitor the progress of cutting operations by sending sensor data using LoRa. The results confirmed the viability of using LoRa

technology in this type of environment, without compromising the reliability of data transmission [21].

In the agriculture field, wireless communication technologies such as LoRa can assist farmers in security, animal and plantation monitoring activities. However, due to the low data rates of LoRa, the transmission of video and image files may be compromised. In [22], the authors propose a visual monitoring system focused on agricultural applications that is based on sending continuous images from a camera using LoRa. A technique is presented in order to optimize the transmission of images via LoRa through which it is possible to decrease the use of bandwidth and improve performance in relation to latency.

A typical LoRa system is composed by two layers: a physical layer and a network layer [23], as can be seen in Fig. 1. The physical layer, conceived and registered by Semtech, focuses on a low-power and long-range wireless communication based on a spread spectrum technique known as Chirp Spread Spectrum (CSS) [24].

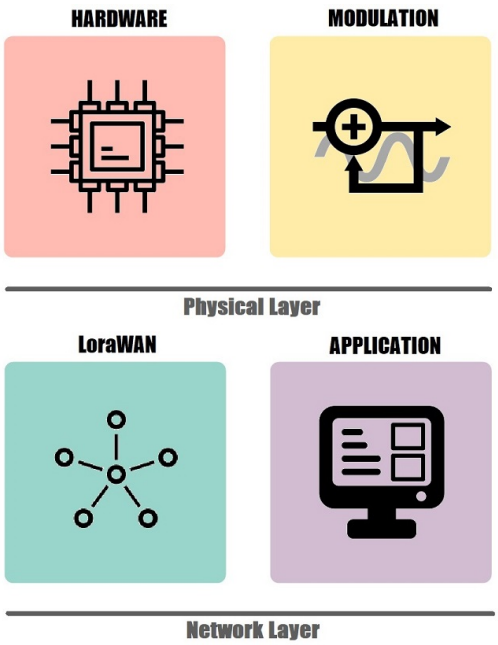


Fig. 1. LoRa's layers of operation.

Based on CSS technique, characterized by its typical spread spectrum, the LoRa modulation is composed of carrier signals named Chirp (Compressed High Intensity Radar Pulse) [23], which are multiplied by the original data signal to attenuate the effects of interference and noise [25]. Thus, the information is encoded according to the position of the frequency discontinuity on each chirp.

The chirp signals have constant amplitude and linearly variable frequency through all the bandwidth. Thus, they can be classified according to the type of frequency variation. The signals for which the frequency runs from a lower frequency f_{low} to a higher frequency f_{high} are known as up-chirps and the opposite, when their frequency initially has a higher value and runs to a lower frequency, characterizes a down-chirp [25].

An experimental example of a LoRa modulated signal can be seen in Fig. 2 which illustrates some important aspects of LoRa modulation, such as the signal bandwidth (BW), the representation of a chirp and also some parts that form a LoRa frame, like the preamble.

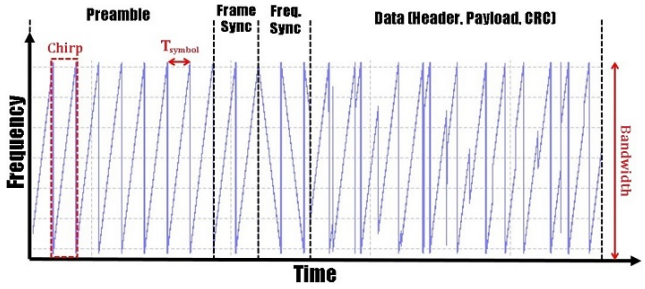


Fig. 2. Example of LoRa Signal.

The modulated signal that is shown in Fig. 2 is formed by several symbols that, together, form a LoRa frame. Each symbol in this LoRa frame corresponds to a single linear chirp and its duration, called symbol time (T_S), can be calculated according to Equation 1 as a function of the Spreading Factor (SF) and the bandwidth (BW) [23], [26]:

$$T_S = \frac{2^{SF}}{BW} \tag{1}$$

In general, the format of a LoRa frame has a defined structure, as is shown in Fig. 2. The LoRa modulated signal initially has a preamble formed by 8 up-chirps followed by 2 up-chirps of frame synchronization (also called sync word), used to differentiate LoRa networks that use the same frequency bands [23], [27]. The next part of the LoRa frame, composed by down-chirps, corresponds to the frequency synchronization symbols, with a total duration of 2 + $\frac{1}{4}$ of symbol [27] and, finally, a part containing an optional header, the data payload and its Cyclic Redundancy Check (CRC) [23].

On the second layer of operation (network layer), LoRa technology is comprised of the network's Medium Acess Control (MAC) protocol, named as LoRaWAN and its Application Server. The LoRaWAN is described by LoRa Alliance as a network protocol optimized for static and mobile devices with lower levels of battery consumption. This network layer is based on a star topology (Fig. 3) with multiple end devices (or nodes) interconnected to multiple gateways, which are connected to the network server. The network server is responsible to pre-process all the network data and send it to the application server. LoRa end devices enable data rates varying from 0.3 kbps to 50 kbps [24].

The gateways are responsible for managing data from one or more end devices, forwarding it to a network server that, in its turn, has the role of centralizing and managing all the system's information, enabling security and error checks and also handling data requests from the application server [28].

LoRaWAN devices are subdivided in three operational classes: A, B and C. Those classes have different data handling methods and capabilities despite being all bidirectional. All LoRaWAN devices must operate as Class A devices and

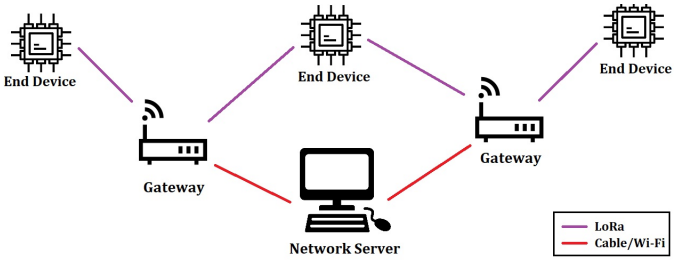


Fig. 3. LoRaWAN network topology.

optionally may also operate as either Class B devices, Class C devices or both [24].

B. Pyroelectric Sensors (PIR)

The Pyroelectric Sensors, also known as Motion Sensors, are a type of Passive Infrared (PIR) Sensor based on the physical principle of pyroelectricity, often used commercially for human motion detection. Its main application is to control automated lamps, though they can also be found on automated toilet flushes and automated doors [29].

The human body is capable of emitting thermal radiation among a wavelength range between 5 μm and 15 μm, in the spectral regions denominated Medium Infrared Spectrum and Long Infrared Spectrum. Those emissions, despite undetectable to the human eye, are utilized by pyroelectric sensors to detect human motion [30].

C. CMOS Image Sensors

Image Sensors are, in general, any kind of device capable of capturing images in the form of coded data that can be used to reconstruct the image in a different medium. These sensors can be constructed either as analog or as digital devices. The fundamental process of digital image sensors is a photoelectric reaction.

The CMOS Image Sensors were developed in the early 1990s and were initially called Active Pixel Sensors (APS), since each picture could be captured and read individually by a sensible element composed of a photodiode and a CMOS acting as a switched capacitor. In these sensors, the typical reading method is called X-Y Reading or Rolling Shutter. Being able to operate each reading cell individually, each pixel is captured and read individually in sequence, element-wise, from left to right and, row-wise, from top to bottom. The whole process lasts only a fraction of a second and consumes small amount of energy as it is based on semiconductors [31].

D. Facial Recognition

Facial Recognition is one of the many biometry techniques currently available. Despite its relatively higher chance of false alarm compared to systems such as iris recognition or fingerprint reading, facial recognition has grown largely on interest over the last few years for presenting a non-invasive method of identity verification [32].

The basis to implement Facial Recognition methods are face detection techniques. Currently, the three most popular methods for face detection algorithms are [32]–[34]:

- Matrix Analysis Methods, such as Cascade classifiers;
- Methods utilizing Histogram of Oriented Gradient (HOG), which are based on Support Vector Machines (SVM) structures;
- Methods utilizing Deep Learning Neural Networks (DNN).

Cascade algorithms were presented in 2001, through the introduction of Viola-Jones' Algorithm, popularly known as Haar Cascade. These algorithms make use of a particular case of the Wavelet Transform, known as Haar Transform or Haar's Wavelet Transform [35].

The HOG algorithms, despite already existing since the 1980s, only became widespread on face detection solutions after Dalal and Triggs [36]. These algorithms use vector fields obtained through the training of SVM structures in order to track silhouettes from objects of interest.

DNN algorithms make use of a more recent and modern technique and are becoming the dominant choice for face detection applications. Those algorithms became popular around 2014, with the release of the library *Deepface*, structured on a convolutional neural network for face detection. The training process in these algorithms is, on average, considerably slower than the previous techniques as it needs a much higher number of training steps. However, the results obtained almost always outclass the other options in terms of precision and accuracy, thus, outweighing its negative aspects [37]–[40].

In this work, facial recognition techniques were implemented based on A. Geitgey's Face Recognition API for python which, on its turn, makes use of face detection models and methods presented on D. E. King's Dlib API/toolkit[1]. This two-layered API-based implementation makes use of a Convolutional Neural Networks (CNN) method for face detection, which is a subtype of the DNN Methods.

1) The face_recognition Module for Python: Created by Adam Geitgey in 2017, this API[2] is an extension made for Python versions 2.7, 3.3 and higher, that can be used either implicitly through coding or directly by command lines [41].

In general terms, it is a modular algorithm, embedded in a python module which is based on facial detection resources contained on Dlib toolkit. Due to that modularization, it presents two major advantages in relation to the default example code for face recognition included on Dlib: Firstly, the Residual Neural Network provided by the API (ResNet-29) is already properly configured and trained, reducing the effort needed to establish a training set of images and to calibrate the network with those images. Additionally, its structure is planned so that each main resource is completely independent from the rest, allowing dependant codes to run almost transparently to what is executed by the API.

2) Dlib: Dlib is an independent and open source toolkit, created in 2002 and developed almost solely on C++, integrating resources that, as stated by its author, provide solutions to several applications [42].

This work indirectly uses the facial detection resource provided by this toolkit. This resource has comparable per-

[1]http://dlib.net
[2]https://github.com/ageitgey/face_recognition

formance to other well established facial detection methods, such as the classic system based on OpenCV library using Haar Cascade methods, such as presented in Viola and Jones (2001) [43].

In terms of architecture, this resource is a Residual Neural Network (ResNet) with 29 layers of convolution, a special case of the DNN method [44]. Its algorithm obtained accuracy levels of 99,38% in validations through widely used benchmarks, such as Labeled Faces in the Wild (LFW) [34].

The detection model provided by the resource presents few downsides when compared to other known models, such as a smaller facial tracking area, usually omitting parts of the forehead and chin. It is also allegedly unable to guarantee proper facial detection for facial areas smaller than 80x80 pixels [45]. On the other hand, this algorithm also presents important advantages in relation to the aforementioned Haar model from OpenCV [45]:

- It has a smaller processing time;
- It is optimized for GPU usage;
- It has a larger tolerance to face obstruction;
- It can detect faces with sharper angles of inclination;
- It is far more robust when detecting false faces.

III. Proposed System Description

The goal of the system prototype is the development of a wireless sensor network able of performing automatic image captures when human presence is detected. The next step is to transfer that image data through routing to a processing center where a facial recognition algorithm looks for faces and identify them based on a pre-stored database, with a certain range of possible applications (such as detecting unauthorized individuals on monitored areas).

The developed system can be defined as a combination of three main elements: the Sensor Nodes (or Transmitters), the Router (or Receiver) and the Processing Unit. It is possible to observe how these elements are disposed on the system in Fig. 4:

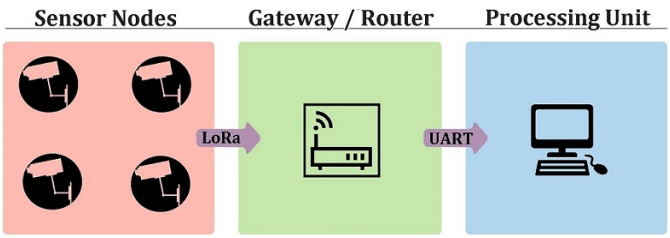


Fig. 4. Diagram of the developed system.

A. Sensor Node

The sensor node is responsible for motion detection, image capture and data transfer. The hardware in this stage is comprised of four devices, as shown in Fig. 5:

1) Image Sensor (Camera): The camera module is the ArduCAM Mini 2MP Plus (Fig. 7(a)), based on the Image Sensor OV2640, capable of capturing images of resolution up to 1600x1200 pixels. The module has an internal image buffer storage and independent buses for command and data transfer based on Inter-Integrated Circuit (I^2C) and Serial Peripheral Interface (SPI) protocols, respectively. Besides, it also has multiple capture modes and an internal routine for compressing images into JPEG format [46].

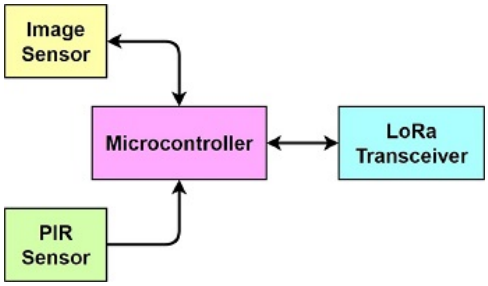


Fig. 5. Components of the developed Sensor Node.

2) Motion Sensor: The motion sensor is the presence detection module HC-SR501 (Fig. 7(b)) [47]. This module is based on the pyroelectric sensor LHI778 [48]. The minimal recovery time for the sensor, which also is the possible time between two detections, is around 7.5 seconds [47].

The sensor has a detection range of up to 7 m over a spread angle of 120 degrees; it can be supplied by a voltage range of 5 to 20 V and produces a logical output of 3.3 V, thus being compatible with many embedded devices.

3) Microcontrolled Unit: The Microcontrolled Unit (MCU) is responsible for the integration process among all the remaining devices. The selected device for this role was the Arduino Mega2560 breadboard (Fig. 7(c)), based on the microcontroller ATMega2560. The board has a 256 kB Flash storage and a SRAM Memory of 8 kB as well as 54 digital pins (of which, 16 are able to generate PWM signals) and 16 analog pins [49]. The algorithm utilized on the transmitter's microcontroller can be seen as a block diagram in Fig. 6.

The natural state of this microcontroller is to remain on an infinite loop, always checking the digital output from the motion sensor. Immediately when a rising edge is detected (logical variation between '0' and '1'), the camera module captures an image, through I^2C bus.

When the capturing process has ended, the module stores the output image, on the output buffer, already compressed in JPEG, ready to be read through the SPI data bus. Before reading this data, the microcontroller enters on a second loop to request the receiver for permission to transmit. This request is sent in the form of the lower byte of the transmitter's LoRa address.

After sending the request, the transmitter waits for the receiver's answer. To accept the request, the receiver must return to the transmitter the same address it had sent. When its request is accepted, the microcontroller begins the process of data transfer from the output buffer of the camera module to the input data buffer from the LoRa module. This transferring process is performed byte by byte, although the LoRa transmission itself is only performed in packets of 256 bytes (or less if the end of the image is achieved).

To deny the request, the receiver must return its own address to the sensor node. In case the request is denied, it will

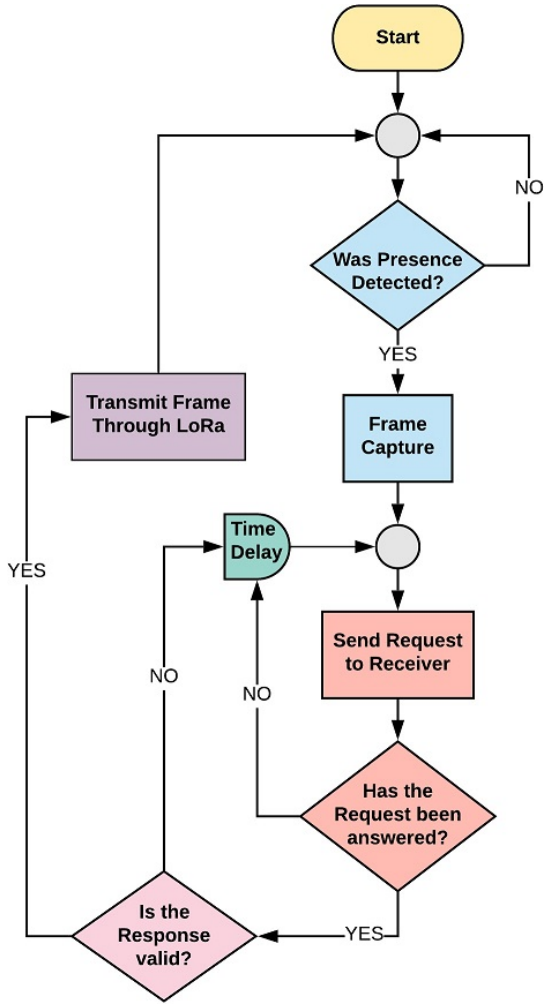


Fig. 6. Block diagram for the algorithm in the Sensor Node.

automatically enters an yield state, promptly discarding the current capture data and returning to an idle state. If three consecutive captures have their requests denied, the node will enter an inactive state on which it will disable its own peripherals and remain inert until a reset is performed on the microcontroller.

When no answer to the request is received after 2 seconds, the transmitter will enter the retry state, in which it will try to send a new request for the same capture data and wait for an incremental delay based on the amount of retrials currently attempted (the first retrial will wait for n seconds before the second retrial which, by its turn, will wait for $2n$ seconds before the third retrial, which will wait for $3n$ seconds before the fourth, and so on). This will be repeated consecutively, until either a request retrial gets accepted or the retry state times out, which will happen after 5 minutes have passed during the retry state.

If the request is accepted on a retrial, the data will be transmitted normally. If the sensor reaches the time out, however, the sensor will enter the yield state, and discard the current data. While the node is on this retry state, similarly as it is on its transmitting state, it will automatically ignore any presence detection. This is done due to hardware restrictions for the node, as it has no secondary buffer to perform image captures in background while handling an already captured image.

When all data is either read from the camera buffer and transmitted or discarded, the empty buffer flag triggers the sensor node to return to its original idle state and starts again to check the motion sensor's output for rising edges.

4) LoRa Transceiver: The communication within the system was based on LoRa modulation. The chosen module for this function is E32-433T20DC (Fig. 7(d)) from ChengDu Ebyte [50]–[52].

This device is based on LoRa chip model SX1278 from Semtech, operating in the frequency of 433 MHz. It performs data transfer through a 3.3 V serial TTL bus, being able to store up to 512 bytes of content in an internal buffer. It is possible to achieve a maximum range of 3 km in direct line-of-sight while still consuming less than 120 mA of current at 20 dBm of power [51].

The module is accessed through seven pins. Two of them are for voltage supply (VCC and GND), one is a status output (AUX), two are for serial data transfer (RXD and TXD) and the remaining are logical inputs for setting the module's operation mode (M0 and M1).

(a) (b) (c) (d)

Fig. 7. Sensor node components **(a)**: ArduCAM Mini 2MP Plus [53]. **(b)**: HC-SR501 [54]. **(c)**: Arduino Mega2560 [49]. **(d)**: E32-433T20DC [50].

B. Router

Considering the three elements of the system (Fig. 4), the router is the simplest one, since its main objective is merely to interface the LoRa network with the processing unit. Thus, three devices are required: a LoRa transceiver, a microcontroller and an Universal Asynchronous Receiver/Transmitter (UART) serial interface, as shown in Fig. 8.

Fig. 8. Components of the Router.

A second E32-433T20DC module was chosen for the LoRa transceiver, acting as receiver. The data reception operates on both universal reading (address `0xFFFF`) to process transmission requests and on point-to-point communication for reading the image data. Besides this software-defined address switching, every other aspect of the receiver module is identical to the transmitter used in the sensor nodes.

For both the microcontroller and UART interface roles, the Arduino UNO breadboard was selected. This choice was made based on the high availability and low learning curve to utilize this type of device. However, any other device with enough digital ports able to support 3.3 V and UART interface could

be selected as a router. The algorithm that was executed for the receiver's microcontroller is shown in Fig. 9.

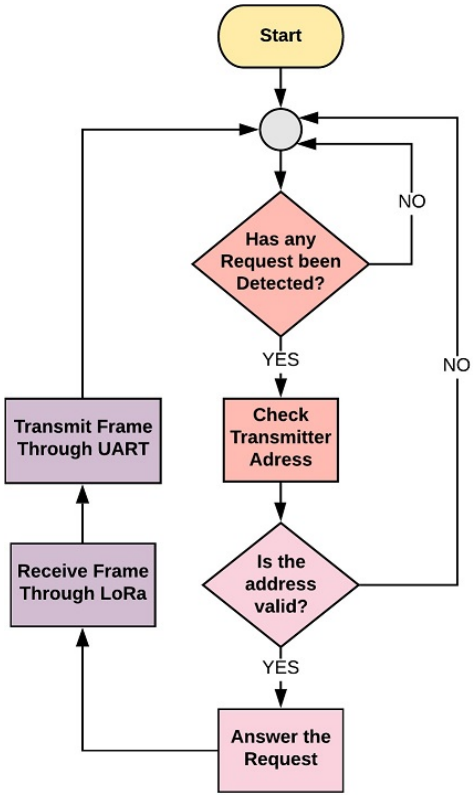


Fig. 9. Block diagram for the algorithm in the Router.

The microcontroller remains at an idle loop in universal reading mode (*broadcast receiver*) awaiting for the first byte reception. This byte must contain a transmission request sent by any transmitter in the network. The request received is both stored and sent to the processing unit, which verifies if the transmitter has sent a valid request (based on a list of registered addresses accessed by the processing unit). Only if the address is valid, the receiver is allowed to authorize this transmitter's request through point-to-point communication by returning its own address. When the transmitter receives its exact address, it is allowed to enter its image transfer stage. The receiver, simultaneously, enters its data reception stage, where it will transfer data byte by byte, from the LoRa module output buffer directly to the processing unit, until the *End of Image* flag is detected by the receiver. When that happens, the algorithm returns back to its initial idle universal reading looping state, awaiting for a new transmission request.

When the router is in idle state and receives a single transmission request, it will always accept it and start receiving data as soon as its request is verified. When it is receiving multiple transmission requests, however, three other situations may occur: The router may receive a transmission request while reading data from another transmission; the router may receive a transmission request while verifying another request; the router may receive, while idle, two or more transmission requests at approximately the same time.

While such feature is a planned upgrade on the development of the prototype, at the moment the router does not handle these situations uniquely and will, in fact, treat them all as the same situation: the router is available for the immediate first request and is busy for every request after it until all data regarding that first request is properly received and relayed to the processing unit. This state of immediately rejecting every other request received is the router's busy state.

To avoid losing incoming transmissions when on a busy state, the router will verify all incoming requests, even those that are rejected by its busy state. In case the rejected request is invalid, the router will answer it with its own address, instead of the requester's, forcing the node to its yield state. When the rejected request is valid, however, the router will not answer it at all, triggering the node's Request Answer Timeout, forcing the node on its retry state, on which it attempts to resend that same request on time-spaced intervals until it is either accepted or it reaches a Request Trial Timeout.

C. Processing Unit

This third element (as shown in Fig. 4) is composed entirely of software and may be implemented in any general purpose computer capable of utilizing image processing and neural network resources. This block diagram is detailed in Fig. 10.

In terms of resources, this algorithm may be splitted in three steps: the serial reading, the file processing and the facial recognition. The main resource utilized on the serial reading step is the *pySerial* module [55]. The file processing step utilizes mainly the *pipeline* resources from POSIX architecture natively incorporated in Python. Lastly, the facial recognition is based on the *face recognition* API [41], [56], [57], which in its turn is based on the face detection library from *Dlib* [42] and on Python's *NumPy* [58] and *PIL* [59] modules.

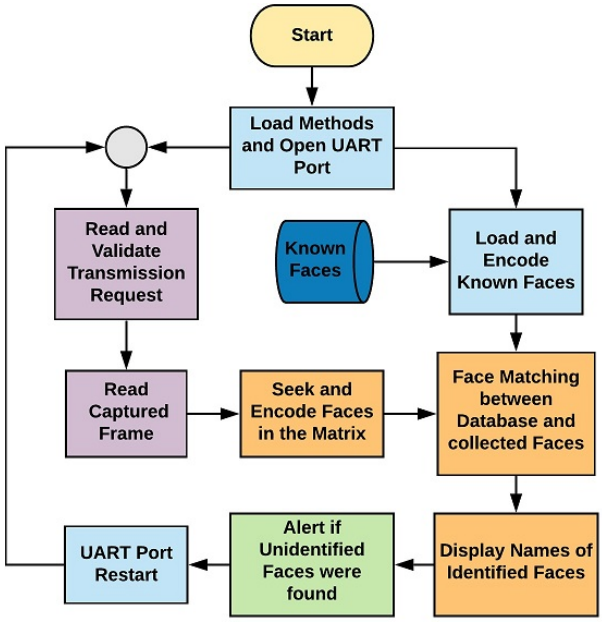


Fig. 10. Block diagram for the algorithm in the Processing Unit.

The sequential execution begins by establishing serial connection with the router's UART interface. Then, the algorithm loads all images present on its face database directory and encodes them into workable data (n-dimensional matrices).

The process of encoding the image is comprised of two steps: Face Locating and Face Description. First, the face locating routine, which is, essentially, a face detecting method, maps the entire image and applies it to either Dlib's HOG facial model or Dlib's ResNet CNN facial model (as both are available). It returns a set of Max-Margin Object Detection (MMOD) Rectangles [60], one for each face detected, which are then converted by the method to a set of four relative coordinates per face, consisting on the four edges of a rectangle that contains a detected face, labeled as a face location.

The next step is the actual encoding of the faces on an image, consisting on mapping the image for the face location provided; ignoring everything outside the face rectangle's borders (which in practice, makes it so that it only maps the inside of said rectangle); and providing a 128-dimensional matrix that objectively and quantitatively describes the facial features inside the provided face location, called a face description. This process of obtaining a face description is iterated for each face location provided and organized as a list of matrices with the same order and number of elements as the list of face locations.

The whole process is performed at the initial stage of the processing unit's routine, before the infinite loop, once for each reference image contained in the database (which contain a single face each) until all images are encoded, as there is no real way to predict which reference image will be needed on each read and recognize iteration before they are in use. One encoding process occurs, however, on each of those iterations, to obtain the potential Face Descriptions on the captured image received through LoRa.

After loading and encoding the database, the algorithm enters in an idle loop awaiting for the first transmission. This state is not infinite, having a configurable timeout value. Whenever this state enters a timeout, the algorithm is aborted. On the other hand, when it detects any amount of incoming serial data, it always treats the first byte as a possible transmission request.

After reading the first incoming byte, it checks if the value is present on a list of registered addresses. In this case, the algorithm allows the receiver to send a positive answer to the requesting transmitter. If not, it will send an error command for the receiver, which will ignore the request.

With the request allowed, the processing unit enters the image reading stage, waiting for the *Start of Image* flag (`0xFFD8`) and then reading all bytes until a *End of Image* flag (`0xFFD9`) is found.

Then, the serial data stored is converted to a *JPEG* file format, then to a 2-dimensional matrix object using the *PIL* module which is encodable by the facial recognition API. The process is exactly the same performed on the reference image files in the algorithm's own database.

After the encoding for both the database and the possible faces on the received image, the next step involves matching the two groups of faces element-wise. The result of this matching is a list of lists containing the values of each individual comparison. Those comparisons are outputted as a non-linear probabilistic value (from 0 to 1) that represents the odds that the two compared faces are associated to the same person. The algorithm then analyses those values, considering that each face on the database is unique, to determine if the given faces from the image are within the known faces from the database.

Faces with no equivalent on the database are labeled as "Unknown" and, according to the proposed routine, are saved as separate image files on a secondary directory, with filenames related to the images of which they were extracted from.

After the facial recognition step, the serial object is reset to flush any undesirable remaining data on the serial buffer and, then, the loop is restarted and the processing unit enters again in an idle finite state awaiting for the next transmission request.

IV. Experimental Results

In this section the results achieved by the proposed system are presented. The goal of the analysis is to evaluate the behavior of certain features of the system and were performed on different stages of the development. According to the development of the system, the results obtained for those experiments are detailed and analyzed.

A. Analysis of Signal Range and Interference in an Urban Scenario

This test consisted on a controlled transmission, at 9.6 kbps data rate and $9600\ baud$, of a single gray scale 240x240 image (Fig. 11). The image was retrieved from an external storage (microSD) connected to the MCU through SPI bus.

Fig. 11. Image used in the test - *lenna240g.jpg* [61]

When initiated, the transmitter would access the image file on the external storage and copy it directly to the input buffer of the LoRa module for transmission.

This process was repeated on a set of 15 different locations along a pre-defined route, maintaining the receiver on the same spot. An external USB power supply was also utilized on this process as a power source.

This test was carried out at the facilities of the Federal University of Paraiba (UFPB) campus, in the city of João Pessoa - Brazil. The router was positioned inside of one of the campus' buildings. There were initially 15 transmission points randomly distributed, labeled sequentially from T01 to T15, reaching up to 713 meters away from the router (on T15). However, due to a signal loss much earlier than T15, the test rout was slightly retraced to a new path, now utilizing the points NT11 to NT15. The traveled route can be observed on

Fig. 12. The red marker represents the location of the router and the green markers indicate the location of the points of transmission.

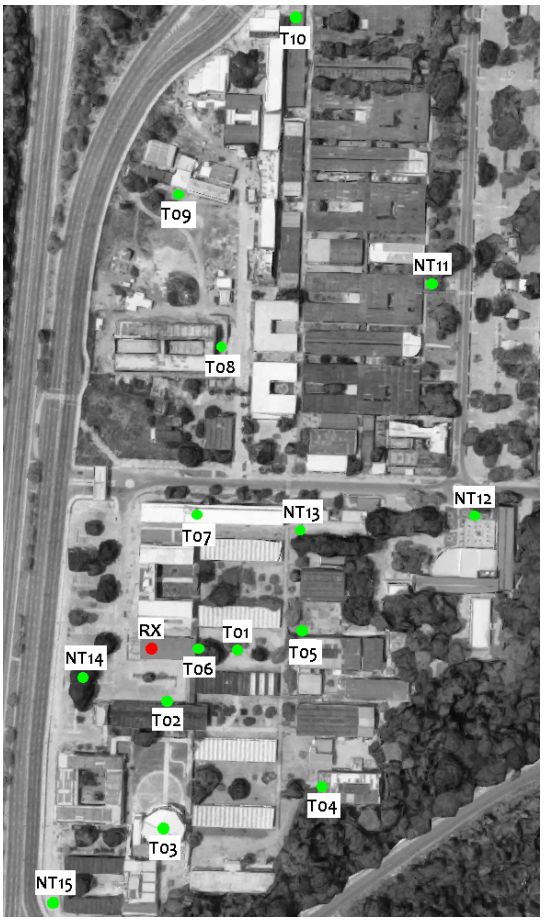


Fig. 12. Map from the region where the test was performed.

The reception time for each transmission point can be observed on Table I. Each point denotes an average of 10 transmission attempts.

TABLE I
DISTANCE AND RESPONSE TIMES ACHIEVED FOR EACH POINT

Point	Distance (m)	Avg. Time (s)
RX	0	34.024
T01	53	100.090
T02	53	75.832
T03	126	50.341
T04	146	Fail
T05	97	78.009
T06	23	62.728
T07	89	Fail
T08	207	Fail
T09	302	Fail
T10	426	Fail
NT11	29	Fail
NT12	223	Fail
NT13	122	129.976
NT14	72	54.694
NT15	191	Fail

According to the manufacturer's documentation, the module's effective range is inversely proportional to the chosen data rate [51]. Being the chosen data rate for this test the second highest, it is reasonable to admit that the real observed range should be much smaller than the nominal range. The experimental values obtained indicate an estimated range between 126 m and 146 m, considering the highly obstructed environment where this test was conducted. Later studies, however, have shown that the system is capable of reaching more than 400 m in direct line-of-sight at its maximum data rate.

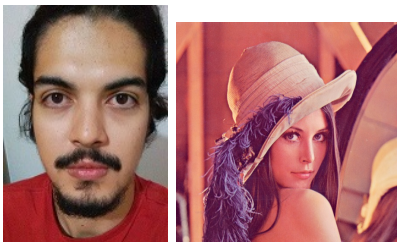
From Table I, it is also noted that there is little or no correlation between the observed reception time at each point and its distance relative to the router. On the other hand, there is an observable influence on time regarding the geographical relative position of each point. Transmissions in direct line-of-sight (T03, T06 and NT13) had the lowest times, being followed by transmissions with mild obstructions (T02 and T05), then by dense vegetation obstructing the path (T01 and NT13) and, finally, by fully obstructed paths (T04 and T07).

Despite showing range values far below its nominal capacity, the results obtained still present a very reliable alternative to other technologies, such as Wi-Fi or Bluetooth. However, the range values may be increased if needed, by replacing the E32 module to a higher power version. The obstruction sensibility may also be softened by selecting paths between each node with less obstacles.

B. Analysis on Identification of Known Faces

The second analysis sought to verify if the developed system would correctly apply the API to identify faces using a database. In order to do that, a two step test was elaborated. This test consisted on evaluating the response of the algorithm on the processing unit when dealing with images containing only "known" faces (registered on the face database). To better evaluate this behavior, the image capturing and transmission would be dismissed, focusing only on the facial recognition part. This was performed by using two test images, stored on a directory in the same machine as the processing unit.

Both steps included a test database containing five image files, and only two of them were utilized (Fig. 13).



(a) (b)

Fig. 13. Test database. **(a)** Vítor. **(b)** Lenna.

In the first step, the test image utilized was "photo collage" of different images from the same individual (Fig. 14), in different conditions. Some pictures containing sunglasses were intentionally selected to test the API's sensibility to partial occlusions caused by facial accessories. When executed, the algorithm detected all 8 faces presented in the image and

correctly labeled them as the same individual in Fig. 13(a). The results obtained were robust and consistent enough to be deemed reliable, being capable to identify even faces from up-scaled images.

Fig. 14. Test image for the First step.

In the second step, the test image consisted of multiple instances of the same image section from *Lenna*, varying in resolution in steps of four times (Fig. 15). This image was used to observe how well the API behaves in relation to small facial elements (obtained either in low resolution pictures or in pictures with faces too far away) and if there was minimum resolution in the detection method. The resolutions used on the image were 240x240, 120x120, 60x60, 30x30 and 15x15 pixels.

Fig. 15. Test image for the Second step.

The results obtained from the processing of this image returned face detections for only three of the five facial elements presented on the picture. That response indicated that the detectability threshold was in a range between 60x60 and 30x30 pixels. To further elaborate this hypothesis, a sub-step was later performed with a third image, produced by one of the facial elements detected by the API in Fig. 13(a). That facial element was re-scaled to explore the best case scenario of the proposed hypothesis (around 30x30), as shown in Fig. 16.

The results shown that, from the three resolutions, only the two largest were detected by the algorithm. This, by itself, can not prove that detectability threshold is exactly 30x30 since factors such as the face orientation, partial occlusions, image quality and other may influence on that sensibility. However, this does show that the threshold is nearby 30x30 pixels (but not necessarily 30x30 pixels) and a conservative approach,

(a) (b) (c)

Fig. 16. Resolutions of a face on the uncertainty threshold. **(a)** 31x31 *px.* **(b)** 30x30 *px.* **(c)** 29x29 *px.*

taking 60x60 or even the 80x80 claimed by the documentation (probably a worst case scenario approach) as the minimum requirement will most likely lead to reliable results.

It is also worth mentioning that the detection threshold issue may be softened or even completely ignored on this specific system due to the hardware limitations of the motion sensor, already restricting a settable maximum range for image captures from 7 m to as low as 3 m away. Further studies should be performed to verify the exact relation between these two thresholds.

C. Analysis on Verification of Unknown Faces

Another test performed on the developed system was to verify if the facial recognition algorithm was capable of detecting the presence of unknown faces on a picture and execute its designated action in response to that detection. The same version of the algorithm from the previous test was utilized here. For the test image, a photography from a technical event hosted at Federal University of Paraiba (UFPB) was used (Fig. 17).

Fig. 17. Image used in the test.

The proposed goal for the system to use facial recognition is that, by recognizing faces on the captured image, it can detect and track unknown faces and, in the presence of one or more faces, initiate a certain pre-determined routine. This pre-determined routine is intentionally left empty to denote that it may change drastically depending on the applications and the intentions of the user. As an example routine, the system saves all tracked unknown faces in a separate directory, with names relative to their original picture.

By running the test image on the algorithm, the processing unit returned six of the nine faces of individuals present on the picture. From the three faces undetected by the API, two were highly obstructed and one was positioned in an angle too sharp to be detected. Both issues are thoroughly described on

the facial recognition's documentation as known limitations for this method.

When executing the example routine proposed, the algorithm created six image files related to the original loaded image file on Fig. 17, where each of them contains one of the six facial elements detected and unidentified. Those files may be observed on Fig. 18, where they share a common filename prefix with the numerical code being the same from the original image and an unique suffix or index, identifying the order on which they were detected by the API.

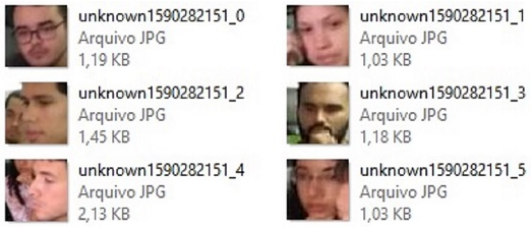


Fig. 18. Unknown faces extracted from the image in Fig. 17.

D. Point-to-Point System Validation

The first system validation was performed addressing the transmission and processing steps considering a point-to-point version of the system (a single transmitter). For that, two pre-defined images were selected and were sent by the transmitter's microcontroller from an SPI external storage module (microSD). The speed settings for the LoRa transmission were 9.6 kbps of air data rate and 9600 bauds.

The first selected image was *lenna240g.jpg* (Fig. 11), here labelled as *L240G.jpg*, previously used on the test described in Section 5.1. The second image was also a gray scale image of 320x240 pixels, labelled *VO_GQVGA.JPG* (Fig. 19).

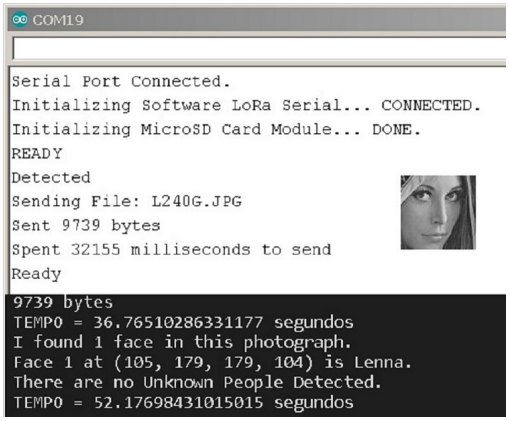
Fig. 19. Second image used in the test - VO_GQVGA.JPG

The first image was used as a reference/control image, since its response was already known. The second image was selected because it had both a known face and an unknown face.

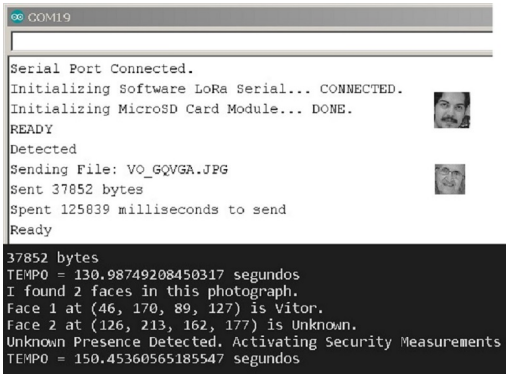
Despite no image captures being actually performed, the motion sensor was still used as a trigger for the transmission event designed for the sensor node.

The results obtained from executing the system for the first test image were summarized on Fig. 20(a). In this image, it is possible to observe the data shown by the transmitter (on the white screen), by the processing unit (on the black screen) and the detected facial element. One can verify that the algorithm correctly identified the evaluated face as Lenna (Fig. 13(b)) among the available database faces.

The response obtained by transmitting the second image, such as in the case of the first, were summarized in a single picture (Fig. 20(b)), with data from the transmitter, the processing unit and the facial elements that were detected. In this case, the known face was correctly identified as "Vitor" (Fig. 13(a)) and the unknown face triggered the example routine.



(a)



(b)

Fig. 20. System return for the transmission of: **(a)**: *L240G.JPG* **(b)**: *VO_GQVGA.JPG*.

After accessing the directory designed for storing unknown faces, it can be seen that a new image file was created regarding the last image received (*VO_GQVGA.JPG*), since it had an unknown face. That image file can be seen in Fig. 21 and is identical to the unknown facial element shown in Fig. 20(b).

Fig. 21. Unknown face extracted from *VO_GQVGA.JPG*

E. Validation of the System working in a LoRa Network

After all previously described tests, only two features of the proposed system would remain to be analyzed: the image capture by the camera module and the network operation itself. Both of these features were verified on this test along with every other feature already analyzed; however, to optimize the performance, the image resolution of the camera was set to a minimum (160x120 pixels).

The goal of this test was to verify if the system would obey to the set of restrictions that were imposed to it in order to manage multiple sensor nodes without conflict issues. These restrictions are:

- A transmitter does not capture a new image until it has sent the current one;
- The receiver ignores any requests from other transmitters while it is on reading mode;
- A transmitter requests transmission until its current image is properly sent.

In order to verify if those conditions were attained, a sequence of transmissions was performed with the intention of verifying all the cases of interaction between two independent sensor nodes:

1) Either A or B node detects motion while the system has entered idle mode for the first time;
2) Either A or B node detects motion while itself is still transmitting;
3) Either A or B node detects motion soon after itself has transmitted;
4) A node detects motion after B node has transmitted;
5) B node detects motion after A node has transmitted;
6) A node detects motion while B node transmits;
7) B node detects motion while A node transmits.

A fragment of one of those sequences performed on this evaluation can be seen in Fig. 22.

It is possible to observe in text each sequential step of the algorithm being executed. The sequence of transmissions was performed repeatedly, varying the order on which the enumerated cases were reproduced. The results have shown that not only the system always obeys its restrictions, but also that each transmission is completely unaffected by previous algorithm runs and previous transmissions.

This means that considering a network with two sensor nodes (A and B) each possible combinational sequence of n transmissions among them is already supported by the system. This can also be extrapolated to a network with m sensor nodes.

The maximum value for m or, in other words, the maximum number of connected nodes on the network, is currently of 255 devices (with valid adresses values between `0x0001` and `0x00FF`), with `0x0000` being reserved for the router. In theory, this limit can go even further in terms of software by using both address bytes and achieving 65533 devices per router, however the hardware restrictions regarding this limitation are unknown and needs verification.

```
Connecting to Serial port COM4 at 57600 baud.
COM4 Connected.

Waiting for the First Transmission (#001)
Detected Request from Device ID 0x00a0.
Request Accepted. Allowing Serial Data Stream (b'TX').
Incoming Serial Data. Commencing Reading Routine.
Serial Reading Complete.
Time Elapsed = 5.798004 seconds
Saving Image as outIMG_1589774819.jpg
Obtaining Face Encodings from "outIMG_1589774819.jpg"
Checking for faces on the Image...
I found 0 faces in this photograph.
Is that a Ghost? 👻 ~~Bo
Time Elapsed = 5.92494 seconds.
Port COM4 Restarted.
Reception Sucessful.

Awaiting a New Transmission. (#002)
Detected Request from Device ID 0x00b0.
Request Accepted. Allowing Serial Data Stream (b'TX').
Incoming Serial Data. Commencing Reading Routine.
Serial Reading Complete.
Time Elapsed = 5.357814 seconds
Saving Image as outIMG_1589774834.jpg
Obtaining Face Encodings from "outIMG_1589774834.jpg"
Checking for faces on the Image...
I found 0 faces in this photograph.
Is that a Ghost? 👻 ~~Boo!
```

Fig. 22. Fragment of a result for the second validation.

V. Discussions, Conclusions and Future Works

Inspired by the rapid growth in the fields of facial recognition and low power long range transmissions, this project was conceived to verify the performance of low-cost remote imaging systems and its feasibility as WISN. Taking into consideration the favorable results on every evaluation performed on the proposed system, it is reasonable to conclude that the project achieved its goal both as an approach of a broad validation of LoRa technology in WISN and as an overall monitoring and identification system.

Regarding the performance of the LoRa module selected for this project (E32-433T20DC), it has shown to be a highly consistent and reliable device for communication on both point-to-point and network scenarios and it presented no issues when dealing with large amounts of data (such as it is required for the kind of application that is a WISN).

In the methodological approach considered for this project, the hardware resources chosen were all commercial devices and the software implementation utilized well-established resources. This approach style also brought a few restrictions to the project. In terms of hardware, the devices, despite being accessible and satisfactory in performance, could not offer the best achievable performance that current embedded technology can provide, which has directly influenced some aspects of the system (such as image capture and quality and transmission speed).

In terms of software, choosing to base the algorithm on a well established API brought better results considering the facial recognition step. However, relying on a third-party algorithm presents, despite in a very reduced scale, more dependency and a potential code conflict, also diminishing the

freedom to precisely adjust the facial recognition method to be optimized to the proposed approach. Fortunately, the positive features of this choice outweighs the potential negative aspects of this decision.

A. Comparison with Related Works

Regarding the performance of our work in terms of the obtained results, five other papers were selected to overall compare their respective findings with our work's results.

TABLE II
OBJECTIVE COMPARISON OF THE RESULTS OBTAINED IN EACH WORK

Paper	Application	Contributions
[13]	Outdoor Surveillance	Vanguard Research, Image Encoder for LPWAN, Transmission Data
[14]	Environmental Monitoring	Image Parsing for LoRa, Vegetation influence on PSNR, Transmission Data
[62]	UAV Surveillance	Compression Method Evaluation
[63]	Agricultural Monitoring	CSMA Mechanism for LoRa
[64]	Site Surveillance	New Network Protocol
This Work	Surveillance/Biometrics	HD Image Parsing for LoRa, Face Recognition for WSN

In terms of similarity, this work is more closely related to Pham (2016) [13] and Jebril et al. (2018) [14] which are, in fact, the main basis on which this research was developed upon. In terms of range in successful transmissions, all three present distinct results which may be justified by differences in hardware, data rate and transmission conditions. However, they have all shown similar behavior regarding the transmission consistency and vegetation packet loss influence.

While Pham [13] focused more on what data was being transmitted, evaluating multiple scenarios regarding image compression and encoding and Jebril et al. [14] had a strong lean on how data was being transmitted, ensuring that the proposed system had a robust and trustworthy transmission, to withstand the adversity of transmitting through a dense tropical forest; this work, following a similar trend, was more engaged on why data was being transmitted, meaning that the application, the part after the image reception, on which face recognition methods were utilized to remotely identify individuals, had a larger role on the development of the research.

B. Future Works and Continuation of this Research

Although the implementation of the proposed system has been successfully accomplished, there is still opportunities to extend and to improve its current state in future works. Some topics of interest that were not properly addressed yet and are potential motivations for subsequent researches are as follows:

- Comparing the hardware components with other equivalents in terms of performance;
- Comparing the utilized DNN API to other available face recognition algorithms and techniques;
- Implementing an original DNN algorithm for face detection in the system and evaluating its performance;
- Evaluating the influence of pre-processing the captured images on the sensor node;
- Embedding the Router and implementing a full network environment;
- Measuring and/or simulating the signal propagation characteristics of the LoRa technology to extend the range of the complete system;
- Comparing the transmission aspects of the LoRa network against alternative technologies;
- Implementing a queuing handler to manage multiple incoming transmissions in real-time.

REFERENCES

[1] K. Ashton. *That 'Internet of Things' Thing*. June 2009. URL: https://www.rfidjournal.com/that-internet-of-things-thing (visited on 08/11/2019).

[2] L. Atzori, A. Iera, and G. Morabito. "The Internet of Things: A Survey". In: *Computer Networks* (Oct. 2010), pp. 2787–2805. DOI: 10.1016/j.comnet.2010.05.010.

[3] K. L. Lueth. *State of the IoT 2018: Number of IoT Devices Now at 7B - Market Accelerating*. IoT-Analytics. 2018. URL: https://iot-analytics.com/state-of-the-iot-update-q1-q2-2018-number-of-iot-devices-now-7b/ (visited on 06/01/2020).

[4] M. Patel, J. Shangkuan, and C. Thomas. *What's new with the Internet of Things?* McKinsey Global Institute. May 2017. URL: https://www.mckinsey.com/industries/semiconductors/our-insights/whats-new-with-the-internet-of-things (visited on 09/01/2019).

[5] K. Mekki, E. Bajic, F. Chaxel, et al. "A Comparative Study of LPWAN Technologies for Large-scale IoT Deployment". In: *ICT Express 5* 5 (Mar. 2019), pp. 1–7. DOI: 10.1016/j.icte.2017.12.005.

[6] Z. Ying S. Lin and K. Zheng. "Design and Implementation of Location and Activity Monitoring System Based on LoRa". In: *KSII Trans. Internet Inf. Syst.* 13 (2019), pp. 1812–1824.

[7] Y. Ma et al. "Development and Application of an Atmospheric Pollutant Monitoring System Based on LoRa-Part I: Design and Reliability Tests". In: *Sensors* 18 (2018), p. 3891. DOI: 10.3390/s18113891.

[8] E. L. Medeiros et al. "Data Acquisition System Development for a Hydraulic Plant using Hybrid Communication Network based on LoRa". In: 2019 IEEE International Instrumentation and Measurement Technology Conference (I2MTC) (2019).

[9] R. Sanchez-Iborra et al. "Tracking and Monitoring System Based on LoRa Technology for Lightweight Boats". In: *Electronics* 8 (2019). DOI: 10.3390/electronics8010015.

[10] H. Zhang, L. Li, and X. Liu. "Development and Test of Manhole Cover Monitoring Device Using LoRa and Accelerometer". In: *IEEE Trans. on Instrum. Meas.* 69 (2020), pp. 2570–2580.

[11] F. O. Ehiagwina, O. O. Kehinde, N. A. Iromini, et al. "Ultra-Low Power Wireless Sensor Networks: Overview of Applications, Design Requirements and Challenges". In: *AJERD* 1 (Dec. 2018), pp. 331–345.

[12] J. M. S. Sant'Anna et al. "Hybrid Coded Replication in LORA Networks". In: *IEEE Trans. Ind. Informat.* 16 (2020), pp. 5577–5585.

[13] C. Pham. "Low-Cost, Low-Power and Long-Range Image Sensor for Visual Surveillance". In: *Proceedings of the 2nd Workshop on Experiences in the Design and Implementation of Smart Objects*. SmartObjects '16. New York City, New York: Association for Computing Machinery, 2016, pp. 35–40. ISBN: 9781450342544. DOI: 10.1145/2980147.2980156. URL: https://doi.org/10.1145/2980147.2980156.

[14] A. H. Jebril, A. Sali, A. Ismail, et al. "Overcoming Limitations of LoRa Physical Layer in Image Transmission". In: *Sensors* 10 (2018), p. 3257.

[15] C. Pham. "Low Cost Wireless Image Sensor Networks for Visual Surveillance and Intrusion Detection Applications". In: *ICNSC 2015 - 2015 IEEE 12th International Conference on Networking, Sensing and Control* (June 2015), pp. 376–381. DOI: 10.1109/ICNSC.2015.7116066.

[16] W. Zhai. "Design of NarrowBand-IoT Oriented Wireless Sensor Network in Urban Smart Parking". In: *International Journal of Online Engineering (iJOE)* 13 (Dec. 2017), p. 116. DOI: 10.3991/ijoe.v13i12.7886.

[17] M. Cerchecci, F. Luti, A. Mecocci, et al. "A Low Power IoT Sensor Node Architecture for Waste Management Within Smart Cities Context". In: *Sensors* 18 (Apr. 2018), p. 1282. DOI: 10.3390/s18041282.

[18] J. Santa, R. Sanchez-Iborra, P. Rodriguez-Rey, et al. "LPWAN-Based Vehicular Monitoring Platform with a Generic IP Network Interface". In: *Sensors* 19 (Jan. 2019), pp. 1–17. DOI: 10.3390/s19020264.

[19] LoRa Alliance. *About LoRa Alliance*. URL: https://lora-alliance.org/about-lora-alliance (visited on 09/04/2019).

[20] M. R. Villarim et al. "LoRa Performance Assessment in Dense Urban and Forest Areas for Environmental Monitoring". In: *INSCIT 2019 - 4th International Symposium on Instrumentation Systems, Circuits and Transducers*. IEEE. ISBN: 9781728121093. DOI: 10.1109/INSCIT.2019.8868567.

[21] L. Tessaro et al. "LoRa Performance in Short Range Industrial Applications". In: *SPEEDAM 2018 - Proceedings: International Symposium on Power Electronics, Electrical Drives, Automation and Motion* (2018), pp. 1089–1094. DOI: 10.1109/SPEEDAM.2018.8445392.

[22] M. Ji et al. "LoRa-based Visual Monitoring Scheme for Agriculture IoT". In: *SAS 2019 - 2019 IEEE Sensors Applications Symposium, Conference Proceedings* (2019). DOI: 10.1109/SAS.2019.8706100.

[23] A. Augustin, J. Yi, T. Clausen, et al. "A Study of LoRa: Long Range & Low Power Networks for the Internet of Things". In: *Sensors* 16 (Oct. 2016), p. 1466. DOI: 10.3390/s16091466.

[24] LoRa Alliance. *White Paper. LoRaWAN^{TM} 1.1 Specification*. 2017.

[25] S. Devalal and A. Karthikeyan. "LoRa Technology-an Overview". In: Iceca (2018), pp. 284–290.

[26] P. Sommer et al. "Low-Power Wide-Area Networks for Industrial Sensing Applications". In: *Proceedings - 2018 IEEE International Conference on Industrial Internet, ICII 2018* Icii (2018), pp. 23–32. DOI: 10.1109/ICII.2018.00011.

[27] C. Bernier et al. "Low Complexity LoRa Frame Synchronization for Ultra-Low Power Software-Defined Radios". In: *IEEE Trans. Commun.* 68.5 (2020), pp. 3140–3152. ISSN: 15580857. DOI: 10.1109/TCOMM.2020.2974464.

[28] W. Dargie and C. Poellabauer. *Fundamentals of Wireless Sensor Networks. Theories and Practices*. 1st ed. Chichester, WS, United Kingdom: Wiley, 2010. Chap. 7, pp. 183–184. ISBN: 978-0-4709-9765-9.

[29] H. Dunsky. *Passive Infrared Sensors: A Brief Overview*. URL: https://www.inhomesafetyguide.org/passive-infrared-sensors-brief-overview/ (visited on 04/03/2019).

[30] J. Fraden. "Pyroelectric Sensors". In: *The Measurement, Instrumentation, and Sensors handbook*. Ed. by J. Webster and H. Eren. 1st ed. Vol. 2. Boca Ratón, FL, USA: CRC Press LLC, 1999. Chap. 32.7, pp. 1073–1081. ISBN: 978-0-8493-8347-2.

[31] I. Takayanagi. "CMOS Image Sensors". In: *Image Sensors and Signal Processing for Digital Still Cameras*. Ed. by J. Nakamura. 4th ed. Boca Ratón, FL, USA: CRC Press LLC, 2006. Chap. 5, pp. 169–173. ISBN: 978-0-8493-3545-7.

[32] Y. Kortli, M. Jridi, A. Al Falou, et al. "Face Recognition Systems: A Survey". In: *Sensors* 20 (Jan. 2020), p. 342. DOI: 10.3390/s20020342.

[33] N. J. Neethu and B. K. Anoop. "A Survey On Face Detection Methods". In: *ICATET* (2014).

[34] University of Massachussetts. *Labeled Faces in the Wild*. URL: http://vis-www.cs.umass.edu/lfw/ (visited on 09/08/2019).

[35] P. Viola and M. Jones. "Rapid Object Detection using a Boosted Cascade of Simple Features". In: *Conference on Computer Vision and Pattern Recognition*. Vol. 1. Feb. 2001, pp. I–511. ISBN: 0-7695-1272-0. DOI: 10.1109/CVPR.2001.990517.

[36] N. Dalal and B. Triggs. "Histograms of Oriented Gradients for Human Detection". In: *International Conference on Computer Vision & Pattern Recognition (InCVPR)*. San Diego, CA, June 2005, pp. 886–893.

[37] S. Balaban. "Deep Learning and Face Recognition: the State of the Art". In: May 2015, 94570B. DOI: 10.1117/12.2181526.

[38] M. Wang and W. Deng. "Deep Face Recognition: A Survey". In: *arXiv* (Apr. 2018).

[39] Z. Zhao, P. Zheng, S. Xu, et al. "Object Detection with Deep Learning: A Review". In: *IEEE Trans. Neural Netw. Learn. Syst.* PP (Jan. 2019), pp. 1–21. DOI: 10.1109/TNNLS.2018.2876865.

[40] W. Wójcik, K. Gromaszek, and M. Junisbekov. "Face Recognition: Issues, Methods and Alternative Applications". In: July 2016. ISBN: 978-953-51-2421-4. DOI: 10.5772/62950.

[41] A. Geitgey. *The world's simplest facial recognition api for Python and the command line. Documentation. Release 1.2.3*. 2017. URL: https://buildmedia.readthedocs.org/media/pdf/face-recognition/latest/face-recognition.pdf (visited on 08/10/2019).

[42] D. E. King. *A toolkit for making real world machine learning and data analysis applications in C++*. 2017. URL: https://github.com/davisking/dlib (visited on 08/08/2019).

[43] P. Viola and M. Jones. "Robust Real-time Object Detection". In: *Second International Workshop on Statistical and Computational Theories of Vision - Modeling, Learning, Computing and Sampling*. Vancouver, Canada, July 2001.

[44] D. E. King. *High Quality Face Recognition with Deep Metric Learning*. 2017. URL: http://blog.dlib.net/2017/02/high-quality-face-recognition-with-deep.html (visited on 08/08/2019).

[45] V. Gupta. *Face Detection – OpenCV, Dlib and Deep Learning (C++ / Python)*. URL: https://www.learnopencv.com/face-detection-opencv-dlib-and-deep-learning-c-python/ (visited on 09/08/2019).

[46] ArduCam. *Arducam Shield Mini 2MP Plus*. 2019. URL: https://www.arducam.com/docs/spi-cameras-for-arduino/hardware/arducam-shield-mini-2mp-plus/ (visited on 05/16/2020).

[47] *HC-SR501 PIR Motion Detector. Datasheet*. MPJA.

[48] *Dual Element Detector. Datasheet: LHI 778*. 1st ed. PerkinElmer. Apr. 2001.

[49] Arduino Company. *Arduino Mega 2560 Rev3*. URL: https://store.arduino.cc/usa/mega-2560-r3 (visited on 08/30/2019).

[50] Chengdu Ebyte Electronic Technology Co. Ltd. *E32-433T20DC*. URL: http://www.ebyte.com/en/product-view-news.aspx?id=130 (visited on 08/30/2019).

[51] *E32-433T20DC 100mW DIP Wireless Module. User Manual*. 1st ed. ChengDu Ebyte. 2017.

[52] *SX1276/77/78/79 - 137MHz to 1020MHz Low Power Long Range Transceiver. Datasheet*. 1st ed. Semtech. Aug. 2016.

[53] ArduCam. *Arducam 2MP Plus OV2640 Mini Module SPI Camera for Arduino UNO Mega2560 Board*. 2020. URL: https://www.arducam.com/product/arducam-2mp-spi-camera-b0067-arduino/ (visited on 05/16/2020).

[54] RoboCore. *Sensor de Presença PIR - HC-SR501*. URL: https://www.robocore.net/loja/sensores/sensor-de-presenca-pir-hc-sr501 (visited on 08/31/2019).

[55] C. Liechti. *PySerial Documentation*. 2015. URL: https://pythonhosted.org/pyserial/ (visited on 07/13/2019).

[56] A. Geitgey. *The world's simplest facial recognition api for Python and the command line*. URL: https://github.com/ageitgey/face_recognition (visited on 08/08/2019).

[57] B. Traversy. *Examples for Python Face Recognition library*. URL: https://github.com/bradtraversy/face_recognition_examples (visited on 08/08/2019).

[58] NumPy Developers. *NumPy*. 2019. URL: https://numpy.org (visited on 08/31/2019).

[59] F. Lundh. *Python Imaging Library*. 2011. URL: http://www.pythonware.com/products/pil/ (visited on 08/31/2019).

[60] D. King. "Max-Margin Object Detection". In: (2015).

[61] Wikipedia / Wikimedia Commons. *Lenna*. URL: https://en.wikipedia.org/wiki/Lenna (visited on 05/29/2019).

[62] R. Kirichek, V. D. Pham, A. Kolechkin, et al. "Transfer of Multimedia Data via LoRa". In: Sept. 2017, pp. 708–720. ISBN: 978-3-319-67379-0. DOI: 10.1007/978-3-319-67380-6_67.

[63] C. Pham. "Robust CSMA for Long-Range LoRa Transmissions with Image Sensing Devices". In: *Wireless Days* (Apr. 2018), pp. 116–112. DOI: 10.1109/WD.2018.8361706.

[64] C. Fan and Q. Ding. "A Novel Wireless Visual Sensor Network Protocol Based on LoRa Modulation". In: *International Journal of Distributed Sensor Networks* 14 (Mar. 2018), p. 155014771876598. DOI: 10.1177/1550147718765980.

Vítor José Costa Rodrigues is currently working towards his master degree in Electrical Engineering at Federal University of Paraíba (UFPB), Brazil, in the field of electronic systems and automation. He is currently researching on wireless image sensor networks. His B.Sc. degree in Electrical Engineering was granted by UFPB in 2020. Vítor Rodrigues's research interests include embedded systems, digital signal processing, wireless sensor networks (WSN) and internet of things (IoT).

Douglas de Farias Medeiros received the degree in Electrical Engineering from the Federal University of Paraíba (UFPB), João Pessoa, Brazil, in 2018. He is currently a master's student at UFPB in the area of electronic systems and automation and he works with routing algorithms applied for wireless sensor networks. His research interests include Internet of Things (IoT), electronic systems, wireless sensor networks, micro-controllers and telecommunication systems.

Fabrício Braga Soares de Carvalho received his D.Sc. degree in 2015, his M.Sc. degree in 2006 and his B.Sc. degree in 2003 from Federal University of Campina Grande (UFCG), in Brazil, all in Electrical Engineering. He is Assistant Professor at the Federal University of Paraíba (UFPB), in Brazil, since 2012. He is the founder and leader of the Communications and Signal Processing Research Group (GCOMPS) at UFPB. He is an IEEE Senior Member. He received the Young Professional Award 2018 from IEEE ComSoc Latin America. He is one of the authors of the book ”Spectrum Sensing Techniques and Applications”, published in 2017 by Momentum Press, in New York, US. He has published more than 100 papers in journals and conferences. His research interests include wireless communications, cognitive radio, spectrum sensing and IoT applications.

Waslon Terllizzie Araújo Lopes was born in Petrolina, Pernambuco, Brazil on December, 29, 1974. He received the B.Sc. and M.Sc. degrees in Electrical Engineering from Federal University of Paraiba, Brazil, in 1998 and 1999, respectively. He received his D.Sc. degree in Electrical Engineering from Federal University of Campina Grande, Brazil in 2003. From August 2003 to December 2009 he was with ÁREA1 College of Science and Technology in Salvador, Brazil, where he was the head of the Telecommunications Group. From November 2018 to October 2019 he was visiting professor in the University of Toronto, Canada. Currently, Dr. Waslon is Associate Professor in the Department of Electrical Engineering of the Center of Alternative and Renewable Energy of Federal University of Paraíba, Brazil. He is also Financial Director of the Institute for Advanced Studies in Communications (Iecom). His research interests include robust vector quantization, wireless communication systems, communication theory, and digital signal processing. Dr. Waslon co-authored the books “Communications, Information and Network Security” published by Kluwer Academic Publishers and “Spectral Sensing: Techniques and Applications” published by Momentum Press. Waslon Terllizzie is member of the Brazilian Telecommunications Society (SBrT) since 1997 and Senior Member of the Institute of Electrical and Electronics Engineers (IEEE).